

Abstract of CN1403932

The machine vision system consists of two parts: hardware system and software system. The hardware system includes a high-speed image input channel comprising analog multiplex switch backing up several video cameras, the first low-pass filter, the first A/D converter and decoder module, the second low-pass filter, and the second A/D converter and decoder module; a DPS system interconnecting with frame memory and video bus expanding interface; a frame memory also connected to an image pretreating module; a bus controller for the image pretreating module to connect with address bus, data bus and control bus; a search list and PCI bus interface; and a computer with RS232 interface. The software system consists of hardware drive program, system platform and image processing and target distinguishing software package.

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06F 13/14

G06F 13/36 G06T 1/00

G06K 9/36



[12] 发明专利申请公开说明书

[21] 申请号 02139470.9

[43] 公开日 2003 年 3 月 19 日

[11] 公开号 CN 1403932A

[22] 申请日 2002.10.16 [21] 申请号 02139470.9

[71] 申请人 西安交通大学

地址 710049 陕西省西安市咸宁路 28 号

[72] 发明人 郑南宁 薛建儒 程 洪

[74] 专利代理机构 西安通大专利代理有限责任公司

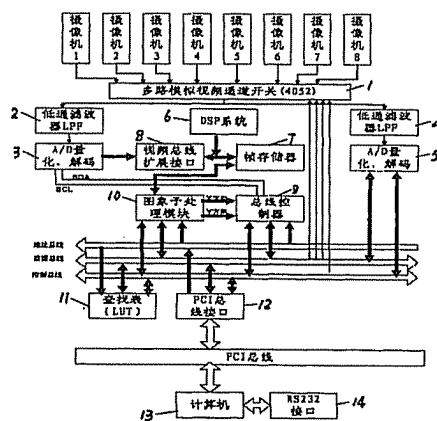
代理人 李郑建

权利要求书 2 页 说明书 24 页 附图 3 页

[54] 发明名称 基于 PCI 和视觉总线的可重构机器视觉系统

[57] 摘要

本发明公开了一种基于 PCI 和视觉总线的可重构机器视觉系统，该视觉系统由硬件系统和软件系统两部分组成；硬件系统包括：可支持多个摄像机的多路模拟通道开关，多路模拟通道开关与第一低通滤波器、第一 A/D 量化、解码模块和第二低通滤波器、第二 A/D 量化、解码模块构成主、副高速图像输入通道；一 DPS 系统，DPS 系统与帧存储器和视频总线扩展接口互连；帧存储器还与图像预处理模块互通；总线控制器、图像预处理模块各自与地址总线、数据总线和控制总线连通；地址总线、数据总线和控制总线还连接有一完成非线性运算的查找表和 PCI 总线接口；PCI 总线接口与 PCI 总线和一计算机连通，计算机带有 SR232 接口；软件系统由硬件驱动程序、系统平台和图像处理目标识别软件包组成。



ISSN 1008-4274

1. 一种基于 PCI 和视觉总线的可重构机器视觉系统, 该视觉系统由硬件系统和软件系统两部分组成; 其特征在于, 硬件系统包括:

一可支持多个摄像机的多路模拟通道开关[1], 多路模拟通道开关[1]与第一低通滤波器[2]、第一 A/D 量化、解码模块[5]和第二低通滤波器[3]、第二 A/D 量化、解码模块[6]构成主、副高速图像输入通道; 第一 A/D 量化、解码模块[5]与一总线控制器[9]相连接; 并与视频总线扩展接口[8]互连;

一数字信号处理系统 (DSP) [6], DSP 系统[6]与帧存储器[7]和视频总线扩展接口[8]互连; 帧存储器[7]还与图像预处理模块[10]互通;

一总线控制器[9]和一图像预处理模块[10], 并各自与地址总线、数据总线和控制总线连通;

地址总线、数据总线和控制总线还连接有一完成非线性运算的查找表[11]和 PCI 总线接口[12]; PCI 总线接口[12]与 PCI 总线连通, PCI 总线与一计算机[13]连通, 计算机[13]带有异步串行通讯 (RS232) 接口[14];

软件系统由三部分组成:

A. 硬件驱动程序

硬件驱动程序主要完成对系统中的硬件部分进行初始化;

B. 系统平台

系统平台采用 WINDOWS98;

C. 图像处理目标识别软件包

图像处理目标识别部分主要提供一些高速图像处理算法软件包, 包括: 图像预处理软件包、图像的边缘检测软件包、图像空间变换软件包、图像特征提取软件包、图像的识别软件包。

2. 根据权利要求 1 所述的基于 PCI 和 Vision 总线的可重构机器视觉系统, 其特征在于, 所述图像处理目标识别软件包包括图像预处理、特征抽取和分割、目标识别三大功能模块;

1) 图像预处理包括的内容有灰度校正、噪声过滤和畸变校正;

2) 特征抽取部分实现边缘检测连接、纹理分析、区域分割算法;

3) 目标识别部分包括的内容有运动目标检测、模板匹配、部分遮挡工件识别;

物体识别的基本方法是建立物体模型, 然后使用各种匹配算法从真实的图像中识别出与物体模型最相似的物体;

物体识别的基本步骤是: 建立模型库、物体特征提取、模版匹配、假设结构的验证等, 相应的识别系统包括四个主要模块: 模型库、特征检测器、假设生成(hypothesis formation)、和假设验证(hypothesis verification);

采用基于模型的工件识别方案, 该方案先把要识别的工件的模型存储起来, 然后通过一种‘猜测-验证’的匹配技术, 快速识别出视野中所需要的工件。

基于 PCI 和视觉总线的可重构机器视觉系统

一、技术领域

本发明属于计算机控制领域，涉及通过模拟人的视觉机理而给予机器人以视觉功能的智能控制，特别涉及基于 PCI 和 Vision 总线的可重构机器视觉系统。

二、背景技术

机器人视觉是一门新兴的发展迅速的学科。通过模拟人的视觉机理而给予机器人以视觉功能已成为目前智能控制领域中的一个重要课题。八十年代以来，机器人视觉的研究已经历了从实验室走向实际应用的发展阶段。从简单的二值图像处理到高分辨率多灰度的图像处理，从一般的二维信息处理到三维视觉机理、模型和算法的研究都取得了很大的进展。而计算机工业水平的飞速提高以及人工智能、并行处理和神经元网络等学科的发展，更促进了机器人视觉系统的实用化和涉足许多复杂视觉过程的研究。目前，在发达的工业国家中，机器人视觉系统正在广泛的应用于视觉检测，机器人的视觉引导和自动化装配领域中。

在机器人的工业应用中，装配机器人对视觉系统有着更高的要求。通过与非视觉传感器（如力觉、接近觉和触觉等）传感器功能的相互协调配合，视觉系统要完成机器人对装配工件的识别定位、及检测功能，使装配机器人实现典型的装配动作，如抓取，插入和拧紧等动作。目前以研究先进的直接驱动的机器人而著称的美国 Adept 公司已推出了基于中分辨率，多灰度的（128 级）的机器人视觉系统 Adept Vision XGS-II，该系统可用于具有显著二维特征的工件装配机器人。此外还有日本的 FANUC 公司用于装配机器人的 WYEV-110 视觉系统，该系统的特点是图像灰度为 256 级，可进行二维或三维物体位置的检测与定位（应用多个摄像机的场合），三菱电机的 LS-212 视觉系统，日本的这类机器人视觉系统多是应用于轿车装配线上。美国 Intelledex 公司用于装配机器人的 TM200 视觉系统等。西安交通大学在“八五”期间为 863-512 主题型号研制项目“精密装配机器人”研制了一套实时的视觉系统，并以推广应用于国防、医学等领域，获得 96 年国家科技进步二等奖。

但是至今国内市场上尚未出现能用于不同场合的机器视觉系统，也无相应的标准和系统集成。申请人经过努力，已成功的掌握了此项关键技术。所提供完整系统不仅能用于机器人环境，也能用于国防、医学等领域，必将推动国民经济的发展。

申请人自 1986 年成立人工智能与机器人研究所以来，主持和承担了国家七·五科技攻关项目（68-5-17）“通用工业视觉系统”的研制工作，目前该系统已形成低、中档系统结构，适用于不同工业应用场合的需要，该项目已在数省推广应用，取得良好的社会效益，并分别获得机电部科技进步一等奖、国家科技进步三等奖。1987 年该所承担了国家教委基金下达的“三维机器视觉技术的研究”的课题。89 年 1 月通过陕西省教委组织的专家技术鉴定会，

评价意见是达到八十年代中期国际先进水平，其三维物体的识别方法具有创造性，其应用成果获国家教委科技进步二等奖。该系统也已应用于引进的 Intelledex 660 智能装配机器人上，完成了一个既有实际的机器人本体又有我们自行研制的计算机视觉系统，为进一步研究与开发机器人视觉系统奠定了良好的实验条件。

近十年来，申请人承担了与计算机视觉有关的其它课题，如河南省国防工办委托研制的“行使车辆牌照自动识别系统”，该系统通过国防科工委组织的技术鉴定。八八年又承担了北京——石家庄高速公路收费站的车型自动识别分类的研制工作，该系统采用双摄像机，应用模式识别与图像处理技术自动分割自然车流，抽取特征自动识别车型与分类，重复精度达 99% 以上。

申请人在开展应用研究的同时，结合研究生的培养，还注意基础理论与方法的研究工作。近年来进行了三维物体的空间定位与识别；运动模糊图像的恢复方法；边缘聚焦及其它边缘检测的新方法；基于形态学的工件识别；基于多层神经网络的图像分割等的研究。上述的研究分别是在国家自然科学基金、国家教委重点基金、博士点基金资助下进行的。部分研究成果已应用到一些横向课题中，有的发表在国内外学术刊物上。其中二维目标的分析方法（形状曲线拟合）获得 1995 年国家发明四等奖。

在自行研制硬件的工作中，除了上面所提到的通用工业视觉系统，八八年初开始进行基于人工神经网络的图像并行分布式处理方法与系统的研究，九二年完成了采用多 DSP 芯片（TMS320C30）构成的并行分布式系统的整机系统设计与试验。这一系统不仅可用于视觉信息的高速处理，而且也可用于多变量控制系统。1993 年承担国家重点科学研究工程，研制出医用立体 X 光数字减影与处理系统。该系统可实现立体数学减影和处理，系统采用双（计算机）体并行的新结构，以及双总线结构，并具有数字端口，支持流水操作，有丰富的图像与图形控制与处理软件。系统具有多个摄像机输入与时钟的控制电路，以及与视觉总线连接的大容量帧存。该项目获得国家教委科技进步一等奖。

1995 年还完成了 512 主体的型号研制任务“精密装配机器人视觉系统”，该系统有多片 A100 构成大窗口卷积器，并采用具有并行处理能力的 i860 完成实时视觉信息处理。系统具有丰富的处理软件，可完成物体的精确定位和部分遮挡物体识别。1995 年以来，该系统已在国防、医学等领域得到推广应用，获得国家科技进步二等奖。

三、发明内容

在上述研究的基础上，本发明提出了一种基于 PCI 和 Vision 总线的可重构机器视觉系统，该视觉系统由硬件系统和软件系统两部分组成；硬件系统包括：

一可支持多个摄像机的多路模拟通道开关，该多路模拟通道开关与第一低通滤波器、第一 A/D 量化、解码模块和第二低通滤波器、第二 A/D 量化、解码模块构成主、副高速图像输

入通道；第一 A/D 量化、解码模块与一总线控制器相连接；并与视频总线扩展接口互连；

一数字信号处理系统(DSP)，DSP 系统与帧存储器和视频总线扩展接口互连；帧存储器还与图像预处理模块互通；

一总线控制器和一图像预处理模块并各自与地址总线、数据总线和控制总线连通；

地址总线、数据总线和控制总线还连接有一完成非线性运算的查找表和 PCI 总线接口；PCI 总线接口与 PCI 总线连通，PCI 总线与一计算机连通，计算机带有异步串行通讯(RS232)接口；

软件系统由三部分组成：

A. 硬件驱动程序

硬件驱动程序主要是指对系统中的硬件部分进行初始化；

B. 系统平台

系统平台采用 WINDOWS98；

C. 图像处理目标识别软件包

图像处理目标识别部分主要提供一些高速图像处理算法软件包，包括，图像预处理软件包、图像的边缘检测软件包、图像空间变换软件包、图像特征提取软件包、图像的识别软件包。

本发明的基于 PCI 和 Vision 总线的可重构机器视觉系统的其他一些特点是：

所述图像处理目标识别软件包包括图像预处理、特征抽取和分割、目标识别三大功能模块。

1) 图像预处理包括的内容有灰度校正、噪声过滤和畸变校正；

2) 特征抽取部分实现边缘检测连接、纹理分析、区域分割算法；

3) 目标识别部分包括的内容有运动目标检测、模板匹配、部分遮挡工件识别；

物体识别的基本方法是建立物体模型，然后使用各种匹配算法从真实的图像中识别出物体模型最相似的物体；

物体识别的基本步骤是：建立模型库、物体特征提取、模版匹配、假设结构的验证等，相应的识别系统包括四个主要模块：模型库、特征检测器、假设生成(hypothesis formation)、和假设验证(hypothesis verification)；

采用基于模型的工件识别方案，该方案先把要识别的工件的模型存储起来(模型训练)，然后通过一种‘猜测-验证’的匹配技术，快速在视野中的识别出所需要的工件。

四、附图说明

图 1 是本发明的视觉系统硬件配置图；

图 2 是本发明的视觉系统的软件配置；

图 3 是本发明的视觉系统配置基本框图；

图 3-3 是本发明的快速高效的匹配算法总体结构框图；

图 3-4 是本发明的 Freeman 的链码图；

图 3-5 本发明采用的多边形逼近示意图；

图 3-6 本发明采用的具有宽度的直线扫描转换使用的刷子示意图；

图 3-7 是本发明的软件系统平面分布图；

图 4 是本发明的视觉系统硬件结构。

五、具体实施方式

依本发明的技术方案，硬件系统包括：

一可支持多个摄像机的多路模拟通道开关 1，多路模拟通道开关 1 与第一低通滤波器 2、第一 A/D 量化、解码模块 5 和第二低通滤波器 3、第二 A/D 量化、解码模块 6 构成主、副高速图像输入通道；第一 A/D 量化、解码模块 5 与一总线控制器 9 相连接；并与视频总线扩展接口 8 互连；

一数字信号处理系统（DSP）6，DSP 系统 6 与帧存储器 7 和视频总线扩展接口 8 互连；帧存储器 7 还与图像预处理模块 10 互通；

一总线控制器 9 和一图像预处理模块 10，并各自与地址总线、数据总线和控制总线连通；地址总线、数据总线和控制总线还连接有一完成非线性运算的查找表 11 和 PCI 总线接口 12；PCI 总线接口 12 与 PCI 总线连通，PCI 总线与一计算机 13 连通，计算机 13 带有异步串行通讯（SR232）接口 14；

系统具有的两路高速流水线图像通道：一路带有可与帧存、DSP 系统（DSP 系统为可选件）无缝连接的视频总线扩展接口和可实时完成边缘检测、滤波、二维卷积运算等于处理的高速流水图像预处理单元，称为主通道。另一路通道经 A/D 量化编码，形成的数据流直接经 PCI 总线输入计算机，称为副通道。可作为实时数字视频处理（如运动图像压缩、运动图像插补等）的图像输入通道。硬件系统中采用了先进的 PCI 总线技术；逻辑控制部分采用了 FPGA 实现；采用了国外成熟的可级联与重构的专用图像处理芯片 A110；同时由于提供 VISION 局部总线，用户可在原基础上进行硬件重构，一定程度上克服了以往同类系统结构固定、功能单一的缺陷，拓宽了应用范围。

软件系统由三部分组成：

A. 硬件驱动程序

硬件驱动程序主要是指对系统中的硬件部分进行初始化；

B. 系统平台

系统平台采用 WINDOWS98；

C. 图像处理目标识别软件包;

图像处理目标识别部分主要提供一些高速图像处理算法软件包, 包括, 图像预处理软件包、图像的边缘检测软件包、图像空间变换软件包、图像特征提取软件包、图像的识别软件包。

用户可根据需要灵活配置, 操作简洁方便。除系统平台采用 WINDOWS98 外, 硬件驱动程序、图像处理目标识别软件包均自行开发。图像处理目标识别软件包具有三大功能模块: 图像预处理、特征抽取和分割、目标识别等。图像预处理包括灰度校正、噪声过滤和畸变校正等内容。特征抽取部分实现了边缘检测连接、纹理分析、区域分割等多种算法。目标识别部分包括运动目标检测、模板匹配、部分遮挡工件识别等内容。此系统目前已成功投入使用, 在简单理想工况下, 平均工件识别速度为 13 件/分, 识别率为 99%。试验证明采用 PCI 总线、VISION BUS 与 FPGA 技术大大增强了通用性、实时性和可重构性。PCI 总线解决了计算机 I/O 数据吞吐量瓶颈问题, 使得即使是 1024x1024x24Bit 高分辨率的彩色图像也可达到 TV 速率 (PAL, 25 帧/秒) 实时处理; FPGA 技术的高速、高集成度及在线现场可编程性, 使得功能模块的可重组成为可能; 引入 VISION BUS 局部总线, 可与帧存板及 DSP 板实现无缝联接, 使得系统架构更为开发。如此系统不仅可软件重构, 也可在线硬件模块重构, 更可以根据实际需要实现系统重构。可适应多种应用环境。

物体识别的基本方法是建立物体模型, 然后使用各种匹配算法从真实的图像中识别出物体模型最相似的物体。物体识别的基本步骤是: 建立模型库、物体特征提取、模版匹配、假设结构的验证等, 相应的识别系统包括四个主要模块: 模型库、特征检测器、假设生成 (hypothesis formation)、和假设验证 (hypothesis verification) 等。在这个视觉系统中, 我们采用了基于模型的工件识别方案。该方案先把要识别的工件的模型存储起来 (模型训练), 然后通过一种 ‘猜测-验证’ 的匹配技术, 快速在视野中的识别出所需要的工件。其中的关键是要能找到一种有效的描述工件形状特征的方法, 以及设计一个快速高效的匹配算法。总体结构框图如图 3-3 所示。

5. 1 模型库的维护

模型库包含所有的已知模型。模型库的信息却决于物体识别方法, 可以定量、定性或函数描述, 也可以是精确的几何曲线面信息。在大多数情况下, 物体的模型是抽象的特征矢量。特征是物体的一种属性, 比如物体的面积、周长、链表、顶点 (Vertex)、角点 (Corner) 等, 特征在描述和识别物体过程中起着十分重要的作用。

在本系统软件中, 所谓的模型库是一个存放模型特征向量的文件。模型库的维护模块, 具体包括: 通过对标准模型的训练, 提取该模型的特征点, 然后将其入库; 删除是将不需要的模型从模型库中删除; 而显示是将该模型库中的模型在结果视类中进行显示, 规定显示图

像的大小是 512 x 512 Pixels。可以使用已有的模型库，也可通过训练建立新的模型库。

5. 2 特征提取与形状匹配

对工件形状特征的描述是基于其轮廓多边形的。这是因为，首先多边形描述足够稳定而又不失局部细节(只要在拟合多边形时注意控制误差)，它一方面滤除了图像数字化时带来的噪声，另一方面又充分保留了工件形状中的微小细节；其次，任何工件的轮廓都可以表示成多边形。这是因为多边形表示本质上还是一种经过较少抽象的较为原始的直观的形状描述方法；再次，多边形表示便于计算机存储和处理，后面我们可以看到这种表示既节省内存又便于进行特征提取和匹配过程中的快速验证。

下面所说的是基于轮廓多边形的特征提取。

我们知道，对于任意一个多边形，只要找到一组具有一定的几何不变性(指经过平移，旋转不变性，这里由于装配的需要，不同比例的相似形状的工件被认为是不同的工件，所以在描述多边形的特征时，有意去掉了比例不变性)的特征，就可以用这些特征代表这个多边形。在识别时，这些特征能够用来对工件进行很有效的粗分类。

5. 2. 1 基于矩的几何不变性特征提取

采用基于矩(*Moment*)的形状特征描述

记 R 为一个闭合有界区域，定义区域 R 的 (p, q) 阶矩 M_{pq} 为：

$$M_{pq} = \iint_R x^p y^q dx dy, p, q = 0, 1, 2, 3, \dots \quad [5-1]$$

进一步， R 的 (p, q) 阶中心矩 u_{pq} 可以表示为，

$$u_{pq} = \iint_R (x - x_c)^p (y - y_c)^q dx dy, p, q = 0, 1, 2, 3, \dots \quad [5-2]$$

这里， $x_c = M_{10}/M_{00}$, $y_c = M_{01}/M_{00}$.

可以验证低于 2 阶的中心矩有下面的计算公式：

$$\begin{aligned} u_{00} &= M_{00}, \\ u_{10} &= 0, \\ u_{01} &= 0, \\ u_{20} &= M_{20} - M_{10}x_c, \\ u_{02} &= M_{02} - M_{01}y_c, \\ u_{11} &= M_{11} - M_{10}y_c. \end{aligned} \quad [5-3]$$

这些中心矩具有平移不变性(*Invariant to translation*);可以对其进行规范化得到具有比例不变性(*Invariant to scaling*)的表示:

$$\eta_{pq}=u_{pq}/u_{00}^r, \quad r=(p+q)/2+1, p+q=2, 3, 4, \dots \quad [5-4]$$

还可以对上述表示做进一步的变换,增加旋转不变性(*Invariant to rotation*):

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \\ \phi_4 &= (\eta_{30} + 3\eta_{12})^2 + (\eta_{03} + 3\eta_{21})^2 \end{aligned} \quad [5-5]$$

必须指出,在装配机器人的应用中要求能够区分大小不同的工件,所以本发明在具体应用时,有意去掉了规范化过程(式[2-4]),也就是去掉了比例不变性。另外,基于性能和有效性两方面的考虑,只选用了 ϕ_1, ϕ_2 和面积,周长等参数(其实面积也是矩特征)来描述工件形状。

5. 2. 2 局部形状特征提取

在有了工件的轮廓多边形表示及其几何不变性特征之后,就能利用这些信息完成非遮掩、粘连情况下的工件识别。但是,在工件之间存在互相遮掩、互相粘连等情况时,单靠上述特征是远远不够的。这是因为它们只包含了工件形状的整体信息,而遮掩、粘连等情况下这些整体信息不复存在。在这种情况下,必须找到关于工件的局部特征信息,因为除非所有的局部信息的被掩盖了,否则我们就能利用这些局部信息作为工件识别的依据。注意到人对工件形状中的尖角、长直线段、弧段,以及弧和直线连接等局部信息非常敏感的事实。本发明提出了描述工件形状局部特征的“角点(CORNER)”的概念。形象地说,CORNER是上述敏感位置附近的一定大小的子多边形。这些子多边形就代表了工件的局部形状特征。所以只要找到这些子多边形及其相应的几何不变性特征,就可以在遮掩、粘连等情况下对识别工件作出位置猜测。

在本发明的视觉系统中,CORNER是靠如下的规则来选择的:

- (1). 该点处的多边形的顶角与 180° 的差别大于一定阈值. 这样的点实际上靠近曲率的极大值点。然而,还有一些点不是曲率的极大值但也很重要。这样的点在下一个规则中考虑。
- (2). 该点处在曲率变化较为剧烈的地方,如直线段与圆弧相交处,两个不同凹向的弧段的连接处(拐点)。
- (3). 在满足规则(1)或(2)的多边形顶点附近,按照一定的半径画圆弧,和多边形的边缘有两个交点,把这两个交点之间的一段轮廓构成的子多边形作为选定的角点(CORNER)标记。
- (4). 把每个 CORNER 对应的子多边形的矩特征作为角点的形状特征。这些特征就标记了工件轮廓多边形在这些局部位置的形状信息。

然而,即使能够利用 CORNER 等局部信息对大部分工件的识别提供很有效的依据,还有一

些工件,其轮廓曲线不存在符合上述规则的 CORNER. 比如圆形工件:这时候就要通过其他途径来处理这样的工件。在本发明的视觉系统中,对于圆形工件作了特殊处理。在轮廓多边形基础上再作一次抽象通过园拟合的办法把圆形工件的几何特征参数(园的半径 r)提取出来,在以后匹配时对被识别的轮廓及其片断作园拟合,比较拟合出的几何特征参数(园的半径 r')与模型园的几何特征参数(如考察 $\text{abs}(r-r')/r$ 的大小),就可作出匹配猜测。推而广之,对轮廓为二次封闭曲线(或其他预知的曲线)的工件,都可以类似的在轮廓多边形基础上再抽象一步去处理。因为多边形表示还不能抓住这样的形状的本质特征。比如园可以看作正 N 边形当 N 趋于无穷大时的特例,而实际上由于数字化的缘故,总能够得到一个有限边数的多边形而将其归类为一般的多边形,没有抓住其本质特征,只要训练模型时提出这些预知曲线的生成控制参数就可以了。

5.3 猜测验证的匹配技术

通过前面的特征提取,得到了关于工件形状的特征向量 $K=[S, P, \phi 1, \phi 2]^T$, 其中 S 是面积, P 是周长, $\phi 1, \phi 2$ 是两个基于二阶矩的具有旋转平移不变性的参数。定义两个形状 $K=[x1, x2, x3, x4]^T$, $K'=[y1, y2, y3, y4]^T$ 之间的距离为:

$$\text{Distance}(K, K') = \max \{\Delta_i\} \quad [5-6]$$

这里, $\Delta_i = \text{abs}(x_i - y_i) / \min(x_i, y_i)$. $i=1 \sim 4$.

当模型工件特征 K_m 与被识别工件特征 K_o 满足:

$$\text{Distance}(K_m, K_o) \leq \text{Thr} \quad [5-7]$$

时,就得到一个模型 K_m 与目标 K_o 相匹配的猜测。这里 Thr 控制着匹配猜测的严格程度,它是一个小于 1 的正实数。当 Thr 给的越小,产生的匹配猜测就越少,参加验证的匹配猜测数目减少,拒识率增大;反之, Thr 越大,产生的匹配猜测就越多,参加验证的匹配猜测数目增加,识别率增大。在本视觉系统中, Thr 取值为 0.15。

为了提高识别结果的可信程度(尤其是在这个视觉系统的特殊应用背景的情形下,可信度显得尤为重要),在给出匹配猜测之后,必须对得到的匹配猜测加以验证。为此,本发明设计了一个快速高效的验证算法:根据每个匹配猜测,我们可以得到模型多边形 P_m 经过一定的几何变换 TR (平移和旋转)之后存在于视野中的假设。这里的几何变换 TR 可由产生匹配猜测的两个特征向量对应的多边形(或子多边形)确定。记 P_o 为对 P_m 施加变换 TR 后的多边形,则 P_o 就是 P_m 存在于视野中的可能位置,对 P_o 与视野中的待识别多边形重叠程度加以验证,就可以在置信程度下肯定或否定这个匹配猜测。

这种验证方法在本视觉系统中如下快速实现：首先，工件形状是基于多边形表示的，而多边形由直线段组成，直线的快速扫描转换有著名的 Bresenham 算法；其次，被识别目标的轮廓在提取出来之后，也要以一定的线宽（验证时的误差就在线宽中得以体现）和灰度重画在内存中的位图 *Bmp* 中，为此本发明也特别设计了简单快速的画一定宽度直线段的算法。注意到 *Bmp* 只需把目标扫描一次就可生成。最后，在验证时，只要沿着 P_0 的扫描转换点在 *Bmp* 中遍历一次（只读不写），并统计遇到的具有目标多边形灰度的点的个数 N_i ， N_i 就是 P_0 与目标重合程度的度量，记多边形的总长为 N ，定义可信度 (Certainty Factor) 为：

$$CF = N_i / N \quad [5-8]$$

显然, $0 \leq CF \leq 1$.

当 CF 满足：

$$CF \leq CL \quad [5-9]$$

时，可以认为该匹配猜测通过了验证。这里 CL 为预先设定的置信级别参数。 CL 的取值，也影响着识别结果。 CL 给的越小，拒识率降低，误识率增大；反之，拒识率增大，而误识率降低，可信度增大。在本发明的视觉系统中， CL 是可以由用户根据情况设定的。

5.3.1 工件识别模块的具体实现

一)、自动阈值

本发明使用了一种基于二次搜索的遗传算法的边缘检测阈值自动选取，可以实时的对图像的边缘检测进行阈值的自动选取和处理。

二)、边缘跟踪及链码表示

边缘跟踪是从图像中抽取形状信息的常用手法，目的是沿着图像的等色区域的边界轮廓进行搜索，将搜索到的边界点的位置记录下来，形成一个边界点列，这个点列就表示该轮廓。

为了记录边缘跟踪后得到的轮廓点列数据，我们采用著名的 Freeman 链码表示法。它不是逐点记录像素坐标，而是记录相邻像素之间连接方向的方法，这种编码有着数据压缩而又方便处理的好处。本发明采用的方向代码示意如图 3-4。

其中像素 P 到其相邻像素的方向按逆时针顺序编码为 $0, 1, 2, \dots, 7$.

一般的边缘跟踪算法是“小虫爬法”，即模仿虫子爬迷宫总靠着一个方向前进的搜索方法。本发明的边缘跟踪算法是从上至下，从左至右顺序扫描工件图像直至找到第一个边缘点后开始进行搜索的。

三)、基于链码表示的矩计算

前面的矩 (*Moments*) 的定义式是基于区域的，直接计算是很低效的。所以，我们推导了基于链码表示的矩计算方法，该方法在链码跟踪时就可以同时进行，由于链码跟踪过程中只在区域的边缘点上进行，所以这种方法比起直接计算效率更高。

具体的方法可以参考表格 3-3.

表 3-3 由链码计算矩

链码	$2\Delta S$	$2\Delta M_x$	$2\Delta M_y$
0	$2Y$	YY	0
1	$2Y-1$	$Y(Y-1)$	$X(X+1)$
2	0	0	XX
3	$-2Y+1$	$-Y(Y-1)$	$X(X-1)$
4	$-2Y$	$-YY$	0
5	$-2Y-1$	$-Y(Y+1)$	$-X(X-1)$
6	0	0	$-XX$
7	$2Y+1$	$Y(Y+1)$	$-X(X+1)$

表 3-3:由链码计算矩(续)

链码	$6\Delta M_{xx}$	$6\Delta M_{yy}$	$12\Delta M_{xy}$
0	$2YYY$	0	$YY(3+6X)$
1	$Y(Y(2Y-3)+2)$	$X(X(2X+3)+2)$	$Y(Y(6X+3)-4)+X(2-6Y)$
2	0	$2XXX$	0
3	$-Y(Y(2Y-3)+2)$	$X(X(2X-3)+2)$	$Y(Y(-6X+3)-4)+X(6Y-2)$
4	$-2YYY$	0	$YY(3-6X)$
5	$-Y(Y(2Y+3)+2)$	$-X(X(2X-3)+2)$	$Y(Y(-6X+3)+4)-X(6Y+2)$
6	0	$-2XXX$	0
7	$Y(Y(2Y+3)+2)$	$-X(X(2X+3)+2)$	$Y(Y(6X+3)+4)+X(6Y-2)$

表 3-3 中 X, Y 是当前链码对应的边缘点坐标, 而 $\Delta S, \Delta M_x, \Delta M_y, \Delta M_{xx}, \Delta M_{xy}, \Delta M_{yy}$ 分别表示每处理一个链码相应的矩 $S, M_x, M_y, M_{xx}, M_{xy}, M_{yy}$ 的增量。沿着边缘跟踪一遍, 在不同的链码方向处根据上表就可以计算出该工件的矩来。对于 X, Y 的计算可以参考下面的表 3-4。

表 3-4:由链码计算坐标 X, Y

链码	0	1	2	3	4	5	6	7
ΔX	1	1	0	-1	-1	-1	0	1
ΔY	0	-1	-1	-1	0	1	1	1

初值 $X=0, Y=0$ 。每遍历一个链码, 则根据链码值 c 对应的 $\Delta X_c, \Delta Y_c$ 来修改 X, Y 即可:

$$X=X+\Delta X_c, Y=Y+\Delta Y_c$$

四)、多边形逼近

一个数字化的区域边界可以用多边形任意精确地逼近。事实上链码表示本身就可以看作一个以所有边缘点为顶点的多边形。在实际应用中, 希望在保证误差的前提下使多边形的边数尽可能的少。有人证明了满足一定误差的最小周长多边形是存在的。有关多边形逼近的方法在许多资料上都能找到。

在本发明的系统中, 我们没有刻意去寻找最小周长逼近, 因为只需要区域边界的满足一定误差的逼近。本发明使用了基于分裂合并思想而设计的逼近算法。

参见图 3-5, 设 AB 是一段边界, C 和 D 是边界上的点。逼近过程就是:

1. 连接 A, B 形成直线 AB, 考察边界上其他点到直线 AB 的距离, 记其中最大距离为 d , 如果 d 满足一定的误差范围, 则认为从 A 到 B 的边界可以用直线 AB 近似; 否则, 记最远点为 C, 从 C 处把边界分为两部分: AC 段和 CB 段。

2. 继续仿照 1 中的办法, 分别考察 AC 段, BC 段是否符合规定的误差, 根据情况决定继续分裂或停止分裂。

3. 经过分裂之后得到的多边形上有可能顶点数还比较多, 这时候可以考察相邻的一系列顶点之间的边缘, 如果它们可以用一段直线近似的话, 保留近似直线的顶点作为这一系列顶点的替代。这个过程就是合并。

上述分裂合并思想最简单的实现是用递归过程实现。为了加快执行速度, 避免递归过程对大堆栈的要求(堆栈越大, 程序可以利用的动态存储空间减少, 效率降低), 本发明精心设计了等价的迭代算法。

五)、基于多边形表示的矩的计算

在有了多边形表示之后, 希望能够进一步得到基于多边形表示的矩的计算, 因为从实现的角度考虑, 这样就只存贮多边形而丢掉工件图像和链码表示, 从而可以节省大量空间; 另外由于多边形表示的简洁性, 可望在计算时间上大大加快。

下面给出基于多边形表示的矩的计算方法:

把多边形看成是一个个由其质心与每两个相邻顶点构成的三角形组成的, 对于每第 i 个三角形, 有:

$$m_{00}^i = \alpha_i = (x_i y_{i+1} - x_{i+1} y_i) / 2$$

$$m_{10}^i = \alpha_i (x_i + x_{i+1}) / 3$$

$$m_{01}^i = \alpha_i (y_i + y_{i+1}) / 3$$

$$m_{20}^i = \alpha_i (x_i^2 + x_i x_{i+1} + x_{i+1}^2) / 6$$

$$m_{02}^i = \alpha_i (y_i^2 + y_i y_{i+1} + y_{i+1}^2) / 6$$

$$m_{11}^i = \alpha_i (2x_i y_i + 2x_i y_{i+1} + x_i y_{i+1} + x_{i+1} y_i) / 12$$

进一步, 对于整个多边形, 有

$$m_{pq} = \sum m_{pq}^i$$

得到 m_{pq} 以后, 就可以按照前面的定义计算特征 ϕ_1, ϕ_2 了。

而且可以看到, 这个算法要比基于定义式直接计算的效率高得多, 它的时间复杂度为 $O(n)$, 这里 n 是多边形顶点的个数。

六)、圆弧拟合

为了处理工件中的圆弧, 所以需要用圆弧拟合的方法在多边形表示的基础上进行更高层次的抽象。有关圆弧拟合的方法也很多, 在本发明的系统中, 采用了基于多边形顶点位置的最小二乘法。

设 $\{(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)\}$ 是拟合圆弧要用到顶点, 这里 $N \geq 2$ 。

拟合圆弧的问题在本系统中有两种不同的情况, 一是给定一组顶点拟合出一个封闭的圆弧; 二是给定一组顶点拟合出具有一定约束条件的圆弧段。拟合中用到的误差准则是使拟合出的圆弧离给定采样点的距离最小, 即:

设 (x_c, y_c, r) 为拟合出的圆弧, (x_c, y_c) 为圆心, r 为半径。

误差 $L^2 = \sum [(x_i - x_c)^2 + (y_i - y_c)^2 - r^2]$, $i = 0, 1, \dots, N$ 。要达到最小。

下面给出具体的计算公式。

(1). 拟合一个完整圆弧

令:

$$Sx = \sum x_i, \quad Sy = \sum y_i,$$

$$Sxy = \sum x_i y_i, \quad Sx^2 = \sum x_i^2, \quad Sy^2 = \sum y_i^2,$$

$$Sx^2 y = \sum x_i^2 y_i, \quad Sxy^2 = \sum x_i y_i^2, \quad Sx^3 = \sum x_i^3, \quad Sy^3 = \sum y_i^3,$$

这里 $i = 0, 1, \dots, N$ 。又记:

$$A1 = 2 \bullet (Sx \bullet Sx - N \bullet Sx^2)$$

$$B1 = 2 \bullet (Sx \bullet Sy - N \bullet Sxy)$$

$$A2 = B1$$

$$B2 = 2 \bullet (Sy \bullet Sy - N \bullet Sy^2)$$

$$C1 = (Sx \bullet Sx^2 - N \bullet Sx^3 + Sx \bullet Sy^2 - N \bullet Sxy^2)$$

$$C2 = (Sy \bullet Sy^2 - N \bullet Sy^3 + Sx^2 \bullet Sy - N \bullet Sx^2 y)$$

则有:

$$xc = (C1 \bullet B2 - C2 \bullet B1) / (A1 \bullet B2 - A2 \bullet B1)$$

$$yc = (A1 \bullet C2 - A2 \bullet C1) / (A1 \bullet B2 - A2 \bullet B1)$$

$$r = [(Sx^2 - 2 \bullet Sx \bullet xc + N \bullet xc^2 + Sy^2 - 2 \bullet Sy \bullet yc + N \bullet yc^2) / N]^{1/2}$$

这样, (xc, yc, r) 就是拟合出的圆。

(2). 拟合一段通过两个端点 (x_0, y_0) , (x_N, y_N) 的圆弧段(要求这两个端点不重合)。

令:

$$A = x_0 - x_N$$

$$B = y_0 - y_N$$

$$C = (x_0 + x_N) \bullet A / 2 + (y_0 + y_N) \bullet B / 2$$

显然, A 和 B 不能同时为 0。

若 $A \neq 0$, 则有:

$$K1 = 2 \bullet C / A, \quad K2 = x_0^2 + y_0^2, \quad K3 = B / A,$$

$$S1 = \sum [x_i^2 + y_i^2 + K1 \bullet (x_0 - x_i) - K2] \bullet [y_0 - y_i + K3 \bullet (x_0 - x_i)], \quad i=1, 2, \dots, N-1$$

$$S2 = 2 \bullet \sum [y_0 - y_i + K3 \bullet (x_0 - x_i)]^2, \quad i=1, 2, \dots, N-1$$

$$yc = S1 / S2$$

$$xc = (C - B \bullet yc) / A$$

类似地, 若 $B \neq 0$, 则有:

$$K1 = 2 \bullet C / B, \quad K2 = x_0^2 + y_0^2, \quad K3 = A / B,$$

$$S1 = \sum [x_i^2 + y_i^2 + K1 \bullet (y_0 - y_i) - K2] \bullet [x_0 - x_i + K3 \bullet (y_0 - y_i)], \quad i=1, 2, \dots, N-1$$

$$S2 = 2 \bullet \sum [x_0 - x_i + K3 \bullet (y_0 - y_i)]^2, \quad i=1, 2, \dots, N-1$$

$$xc = S1 / S2$$

$$yc = (C - A \bullet xc) / B$$

同时, 有:

$$r = [(x_0 - xc)^2 + (y_0 - yc)^2]^{1/2} = [(x_N - xc)^2 + (y_N - yc)^2]^{1/2}$$

这里, 是基于轮廓多边形的顶点做圆弧拟合的, 使用上述与顶点位置而不是出现频率相关的最小二乘法能够获得很好的效果。另一方面, 这种方法的效率也很高, 因为它最多只需把全部顶点扫描一遍。

七)、用于验证的快速直线扫描转换

直线扫描转换是光栅图形显示学的基本问题。在本发明的系统中, 为了验证匹配猜测, 我们在内存中开辟了一个大小为 768x512 的位图 *Bmp*。在 *Bmp* 中画有被识别目标的轮廓线。

在得出匹配猜测之后，把模型多边形扫描转换在猜测位置上，以得到模型与被识别目标轮廓重合的程度，达到验证的目的。

在验证时，要允许猜测位置有一定的误差。为了把验证误差控制在一定范围内，本发明在转换轮廓线时赋予这些轮廓线相应的宽度，这就需要在扫描转换时考虑线宽的处理。

(1). 直线扫描转换的 Bresenham 法

目前常用的快速直线扫描转换算法是由 Bresenham 提出的。该方法基于整数加法和乘法（可以用移位操作代替），所以速度非常快。

下面是算法的 C 语言表示。

```
void      BresenhamLine(int xs, int ys, int xe, int ye) {
    int    dx, dy, x, y;
    char    IC;
    int    e, s1, s2, i;

    x=xs;      y=ys;      /* Initialize variables */
    dx=xe-xs;   dy=ye-ys;
    dx=abs(dx);  dy=abs(dy);
    s1=(dx>0)?1:(dx)?-1:0;
    s2=(dy>0)?1:(dy)?-1:0;

    IC=0;      /* Interchange dx, dy so that dx>dy */
    if(dy>dx) {
        e=dy, dy=dx, dx=e;
        IC=1;
    }

    e=2*dy-dx;      /* Begin of plotting */
    for(i=0; i<dx; i++) {
        ScanPixel(x, y);      /* plot at (x, y) */
        while(e>=0) {
            if(IC) x+=s1;
            else y+=s2;
            e-=2*dx;
        }
    }
}
```

```

    }
    if(IC) y+=s2;
    else x+=s1;
    e+=2*dy;
}
}

```

在该算法中, *ScanPixel(x, y)*过程具体负责在像素 (x, y) 处的扫描转换。在本系统中, 当扫描转换模型的猜测位置时它负责完成对模型与被识别目标重合程度的计算; 而在扫描转换被识别目标多边形时, 它负责在 *Bmp* 中 (x, y) 位置处画一定宽度的点。

(2). 线宽的处理

生成具有一定宽度直线的方法也有很多。最简单、高效的方法是沿着直线扫描转换的轨迹移动一个具有一定宽度的刷子。刷子的形状通常采取线条形或正方形。

线条形刷子的原理是, 当直线斜率在 $[-1, 1]$ 之间时, 把刷子置为垂直方向, 刷子的中心与直线扫描转换点重合从一端移动到另一端; 否则, 把刷子置为水平方向即可。这种方法的问题是直线的两端总是水平或垂直的, 当比较接近水平的线与比较接近垂直的线汇合时, 在汇合处会出现一个缺口。参见图 3-6(a)。

正方形刷子的原理是, 把刷子的中心点与直线扫描转换点重合从一端移动到另一端就可。但是, 这种方法效率较差, 因为相邻两个扫描转换点附近的刷子会重叠, 重叠区域的像素被刷了两次。参见图 3-6(b)。

在本系统中结合了以上两种刷子。办法是, 在直线的端点处采用正方形刷子, 而在内部其他点处则采用线条形刷子。参见图 3-6(c)。

这种方法, 既保持了线条形刷子的高效性, 又克服了其端点处的缺点。

5. 4 软件的具体实现

该软件系统采用面向对象编程(OOP)和 C++ 类的编程技术, 应用类的封装性特点主要实现了图像预处理类和工件识别类, 来实现上面所说的图像处理和工件识别技术。该软件涉及到了 VC++ 中的多视窗分割技术、动态连接库(DLL)的调用、Microsoft 的 VFW(Video for Windows) 技术等多项软件开发技术。

5. 4. 1 多视窗分割技术

本软件系统面板采用多视窗分割技术, 将整个显示屏幕分成了三个部分。分别是系统控制部分、实时监视窗口部分、和处理结果显示部分。这三部分具体由三个视类组成来实现, 它们分别是控制视 (ControlView)、捕获视 (CaptureView)、和结果视 (ResultView)。它们

的分布如图 3-7 所示。控制视放置系统的一些控制组件（按钮、列表等），完成对系统的功能控制；捕获视上放置视频窗，完成对整个系统的监控；结果视用于显示结果。

5. 4. 2 VFW 技术

VFW(Microsoft Video for Window)是微软公司为开发基于 Windows 的视频捕捉、视频编辑和视频播放等视频应用程序提供的软件开发工具。该开发工具包含了开发视频应用程序所需要的接口函数 (APIs)，以求简化程序员开发基于 Windows 的视频应用程序的工作。

VFW 包括 AVICAP、MCIavi、DRAWDB、AVIFILE、ICM 等多个组件，通过这些组件间的协调工作，来完成视频图像的捕获、播放、编辑、文件管理等各种功能。其中的 AVICAP 子集是用来开发视频捕捉应用程序的，所以该软件也就用 AVICAP 子集来完成了该系统的视频捕获功能。AVICAP 子集包含了一个 AVICAP 窗口类、与视频捕捉相关的回调函数、向 AVICAP 窗口发送消息命令的消息宏函数集。

AVICAP 窗口类是 VFW 定义的 Windows 窗口类，该子窗口的客户区用来显示采集卡传入计算机的实时视频图像。客户区里的图像可以有两种显示模式：一种是覆盖 (OverLay) 模式；另一种是预览 (PreView) 模式。覆盖模式的显示速度比预览模式要快，但是如果应用程序要对捕获的图像进行加工，就只能在预览模式，先把图像送到系统内存，在内存里对图像进行加工，然后再把图像显示出来。下面通过创建一 AVICAP 窗口来说明其具体的应用。

创建 AVICAP 窗口

Windows 的 SDI 应用程序可以在一个框架窗口中显示一个文档窗口。MFC 中，文档窗口包括一个视类，一个文档类，视类只用作与用户交互的接口界面。这样可以将 AVICAP 类窗口定义为视类的子窗口，在窗口客户区中显示采集的视频图像，同时接受用户的命令。由于采用多视窗分割技术，该软件中有一个文档类和三个视类。我们是在视类 CaptureView 中实现 AVICAP 窗口的。

窗口的创建过程如下（在 CaptureView 是类的 OnCreate() 中）：

```
capCreateCaptureWindow(); 创建 AVICAP 窗口
if(capDriverConnect())    AVICAP 窗口与驱动程序连接
capDriverGetCaps();
capSetCallbackOnXXX();    设置回调函数
```

capCreateCaptureWindow() 是 AVICAP 窗口创建函数。capDriverConnect() 将 AVICAP 函数与采集卡的驱动程序连接起来，如果连接成功该函数返回真值。然后调用 capDriverGetCaps() 检查视频采集卡所支持的功能，返回值保存在 CAPDRIVERCAPS {...} 结构中。该结构里的变量表明采集卡能否支持覆盖显示模式、能否改变视频图像的采集硬件、能否改变采集的图像格式等。

capSetCallbackOnXXX()将程序员定义的回调函数通知底层驱动程序,供采集硬件发出中断消息时调用。AVICAP中定义了多个回调函数的形式来响应不同的中断消息,它们通过相似的消息宏函数设置。设置函数的参数中包含待定义的回调函数的函数指针、应用程序的句柄等。

以上是一个视频处理程序最基本部分,当与视频输入设备连接成功后即可对采集的视频流进行处理。如捕获、设置显示模式、设置视频源对话框等。

5.4.3 图像预处理部分

一般情况下,成像系统获取的图像(即原始图像)由于受到种种条件限制和随机干扰,往往不能在视觉系统中直接使用,必须在视觉的早期阶段对原始图像进行灰度校正、噪声过滤等图像预处理。对机器视觉系统来说,所用的图像预处理方法并不考虑图像降质原因,只将图像中感兴趣的特征有选择的突出,衰减其不需要的特征,故预处理后的图像输出图像并不需要去逼近原图像。在该软件中,图像预处理部分主要实现一些经典的算法,如平滑降噪算法、直方图均衡等。具体实现的类是 CimagePreProcess Class。其处理函数接口如下所示。

一)、平滑降噪算法

图像常被强度随机信号(也称噪声)所污染。一些常见的噪声有椒盐(Salt & Pepper)噪声、脉冲噪声、高斯噪声等。椒盐噪声含有随机出现的黑白亮度值。而脉冲噪声则含有随机的白强度值(正脉冲噪声)或黑强度值(负脉冲噪声)。针对噪声的这些特点,实现了如下的算法。

(1). 中值滤波

void MidErase(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

函数功能: 对一帧图像进行中值滤波

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height: 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(2). 均值滤波

void AverageFilter(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

函数功能: 对一帧图像进行均值滤波

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(3). 高斯滤波

```
void Gaussian(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height, double StdError);
```

函数功能: 对一帧图像进行高斯滤波

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height: 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

二)、边缘提取算法

图像分析和理解的第一步常常是边缘检测 (edge detection)。而该视觉系统正是基于边缘进行特征提取, 而后进行识别的。实现的算法及接口函数如下所示。

(1). Sobel 算子

```
void Sobel(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)
```

函数功能: 对一帧图像用 Sobel 算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height: 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(2). Prewitt 算子

```
void Prewitt(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)
```

函数功能: 对一帧图像用 Prewitt 算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(3). Laplacian 算子

```
void Laplacian(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)
```

函数功能: 对一帧图像用 Laplacian 算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(4). 梯度算子 (roberts)

void Gradient(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

函数功能: 对一帧图像用梯度算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(5). Marr 算子

void Marr(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

函数功能: 对一帧图像用 Marr 算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

(6). Kirsch 算子

void Kirsch(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

函数功能: 对一帧图像用 Kirsch 算子进行边缘提取

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

三)、图像直方图

图像直方图说明了一幅图像的灰度分布情况, 是图像分析中一种很简单和很有用的工具。而直方图均衡则是图像增强的一种有效方法。其函数接口如下。

(1). 求取直方图

void Histogram(unsigned char *SrcImg, double DestData[256], int Width, int Height)

函数功能: 求一帧图像的直方图

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: double DestData[256]: 存放图像的直方图, 下标表示灰度值, 数组

表示具有此灰度值的像素的个数

(2). 直方图均衡

```
void HistogramBalance(unsigned char *SrcImg, unsigned char *DestImg, int
Width, int Height)
```

函数功能: 用直方图度图像进行增强

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

四)、图像二值化

由于计算二直图像特性的算法非常简单, 容易理解和实现, 并且计算速度很快, 再加上二值视觉所需的内存小, 对计算设备要求低等因素, 所以, 二值视觉系统在机器视觉中是十分有用的。该软件实现了如下的二值化算法。

(1). 简单二值化

```
void BinaryImg(unsigned char *SrcImg, int *SignImg, unsigned char
*DestImg, int Width, int Height, int Thd)
```

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

int Thd: 所取阈值大小

出口参数: int *SignImg: 为处理完之后图像的符号表示, 其中 1 表示物体, 0 表示背景

unsigned char *DestImg: 为处理完之后的图像数据

(2). 自动阈值二值化

```
int AutoThresh2(unsigned char *SrcImg, int Width, int Height)
```

```
int AutoThresh(unsigned char *SrcImg, int Width, int Height)
```

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: int: 返回所需的阈值

五)、细化

细化(thinning)是一种图像处理运算,可以把二值图像区域缩成线条,以逼近区域的中心线,也称之为骨架和核线。细化的目的是减少图像成分,直到只留下区域的最基本信息,以便进一步分析和识别。其函数接口如下。

Thinking(unsigned char *SrcImg, unsigned char *DestImg, int Width, int Height)

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

int Width : 图像帧的宽度

int Height : 图像帧的高度

出口参数: unsigned char *DestImg: 为处理完之后的图像数据

5. 4. 4 工件识别部分

该部分主要完成工件的识别,具体实现的类是 CPreRecognition Class。它实现了基于边缘的链码算法,角点、顶点等特征提取算法,多边形拟和算法,匹配猜测算法,及模型库的维护等功能。其主要函数接口如下,具体清参看源程序代码。

全局变量定义在 PRecognition.h 头文件中。

(1). void FindObjects(unsigned char *SrcImg)

函数功能: 通过链码寻找图像中的物体

入口参数: unsigned char *SrcImg: 为将要处理的一帧图像的像素

输出: 如果找到符合要求的物体则弹出一提示框“Normal exit F0!”,否则会弹出一些寻找失败的信息。

(2). void Fit(POLY *pp)

函数功能: 通过顶点来拟合多边形

入口参数: POLY *pp: 要拟合的多边形

输出: 如果成功回弹出一提示框显示多边形的一些参数,否则会出现一些失败信息。

(3). int LocatePoly(POLY *mp, POLY *op, unsigned char *img)

函数功能: 模型库中多边形 mp 和目标多边形 op 进行匹配

入口参数: POLY *mp: 模型库中的多边形

POLY *op: 目标多边形

unsigned char *img: 用于验证的图像缓存

输出: 如果匹配成功将显示一消息框报告成功,否则返回值为零。

(4). int LoadModelFile()

函数功能: 装入模型库

输出: 如果装入失败回显示相应的消息框指示装入失败; 否则就返回 1

5. 5 本发明的特点是:

1) 研制开发了具有多台摄像机的多路切换电路的视频输入与 A/D 单元以及所需的外部触发电路。根据视频总线设计专用的流水线计算单元（如直方图统计、双阈值二值化、二值图像的面积、矩等基本参数的计算）以及扩充其它专用功能模块。

实现的主要功能如下：

- 实时图像采集功能：

视频输入信号 CCIR、RS170 标准、隔行或逐行扫描。

采集速度 25 帧/秒。

图像分辨率 512 X 512 X 8 bit

图像板内存储 1024 X 1024 X 16 bit

与视频总线相连的图像加速板 IPA 及专用的流水线计算单元。

视觉系统总线：PCI 总线。

- 多摄像机输入：

可同时支持 8 个物理像机，每个物理像机又可对应 4 个虚拟像机。

对于不同的像机安装方式，固定和移动提供相应的视觉校正功能。

像机对（16mm，25mm，50mm，75mm）镜头任选。

- 具有与机器人的装配作业机传送带节拍（间歇式）或慢速（6-12 件/分）相适应的实时检测和识别功能。传送带的状态可内外触发电路和串行接口的通讯来与视觉系统同步。

- 视觉系统与机器人控制器的通讯只传输识别结果或某些参数，而不是图像数据，因而采用 RS-232 借口，使系统在现场工作可靠。

2) 以 C++语言和机器汇编语言为主。编制视觉系统的识别软件。

其中包括：

- 编制允许有一定的接触或遮挡的具有比例（Scale），旋转（Rotation）及位移（Translation）不变性的二维工件识别软件。对于几何边数小于 30，孔数小于 10 的工件实时提取多类基于总体和局部的特征。主要的基本特征参数有：

- 面积、周长

- 孔数及几何布局

- 几何中心及重力中心

- 主轴方向

几轴与重力的轴的方向差

- 紧凑度（compact index）

- 长度特征（enlongation index）等

并在以上基本特征基础之上

- 编制基于区域模板匹配的实时工件定位和识别软件以及开发数字形态学的实时处理软件。

- 编制基于梯度图像的实时模板匹配算法软件。

- 研究一种基于CAD模型和视觉感知组合（visual perception organization）基础之上的单三维物体的动态识别和三维定位算法。在具体应用过程中，通过系统开关和参数设置功能选择不同应用场合下的优化识别方案，对于一般复杂程度的工件具有1-10工件/秒的识别速度，平面定位精度误差不大于1像素，识别率大于96%，拒识率大于99%。

3) 以C++语言为基础，编制视觉系统的系统软件和各种辅助软件。

其中包括：

- 具有图形（线条、圆弧等几何图形），开窗（windows），标尺（Rulers）。图像放大（Zoom）、旋转（Rotation）和平移（Pan）等编辑显示功能。

- 编制方便迅速的视觉信息管理软件，有效的安排在线、离线数据库的使用。

- 编制对摄像机进行校正的支持软件。采用方便的人机交互方式，结合精度较高的测试工具，完成图像像素长宽比、物理尺寸几何校正，计算摄像机的有效焦距、图像中心及相对主参考系的旋转、平移等参数。

4) 光照条件：针对装配机器人的特点，允许采用背透光、结构光等特殊光源，可采用光强稳定的、漫射的和无阴影的荧光灯光源。

5. 6 本发明的系统所达到的一些性能指标如下：

1) 硬件方面

①实时图像采集

- 支持多摄像机输入（可同时支持8个物理像机，每个物理像机又可对应4个虚拟像机），对于不同的像机安装方式提供相应的视觉校正功能。

- 视频输入信号CCIR、RS170标准、隔行或逐行扫描。

- 采集速度25帧/秒（CCIR），或30帧/秒（RS170）。

- 图像分辨率512 X 512 X 8 bit。

- 图像板内存储1024 X 1024 X 16 bit。

- 支持PAL/NTSC/SECAM制式模拟输入，能接收复合视频和S-Video信号输入。

- 支持YUV、RGB、YcrCb多种格式输出。

- 与视频总线相连的图像加速板IPA及专用的流水线计算单元。

- 支持图像窗口平滑伸缩和任意剪裁。

②实时图像处理

- 与专用视频总线相连。

- 对 512 x 512 x 8 bit 的图像, 在 X 和 Y 方向上能分别完成最大窗口为 6 x 7 的二维卷积运算, 可实现线性平滑去噪、边缘提取、边缘增强及二值图像细化等各种典型低层次处理算法。

- 具有微机接口, 主机可动态改变控制寄存器和系数寄存器参数。

- 高速 RAM 构成外部 LUT 表, 主机可加载不同内容, 实现复杂非线性变换。

③总线规范

- 零等待 PCI 总线接口。

- 可扩展的视频总线接口, 能直接输出数字视频。

④装配控制

- 具有与机器人装配作业及传送带节拍(间歇式)或慢速(6-12 件/分)相适应的实时检测和识别功能。传送带的状态可通过串行接口的通讯来与视觉系统同步。

- 视觉系统与机器人控制器的通讯采用 RS-232 借口, 仅传输识别结果和控制参数。

2) 软件方面

①驱动程序

- 基于 WINDOWS98 平台, 适于移植到网络环境。

低层次图像处理部分

- 简单理想工况下, 通过填写寄存器中适当的值, 用硬件实现实时处理。复杂背景弱噪声下, 用主机实现近实时处理。

②高层次图像识别部分

能识别有一定的接触或遮挡的工件, 对检测识别的对象具有比例旋转及位移不变性。对于几何边数小于 30, 孔数小于 10 的工件, 可实时提取多类基于总体或局部的特征。识别时利用的主要基本参数是: 面积、周长、几何中心、主轴方向和孔数(及几何布局)等。

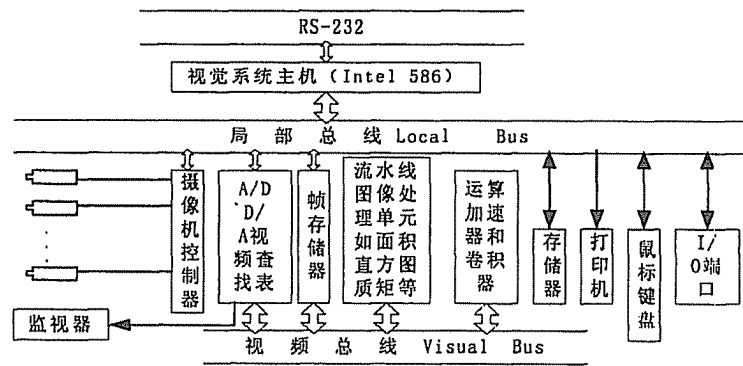


图 1

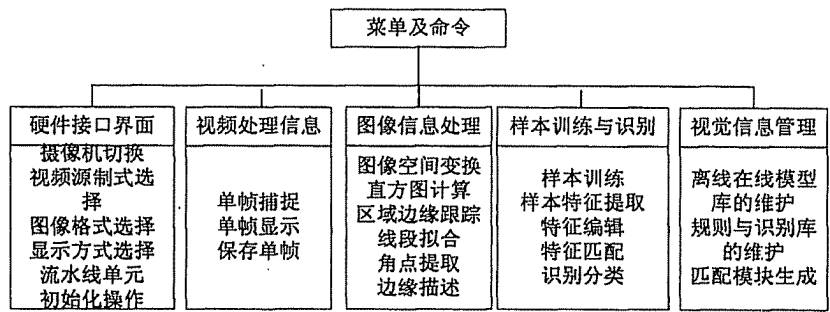


图 2

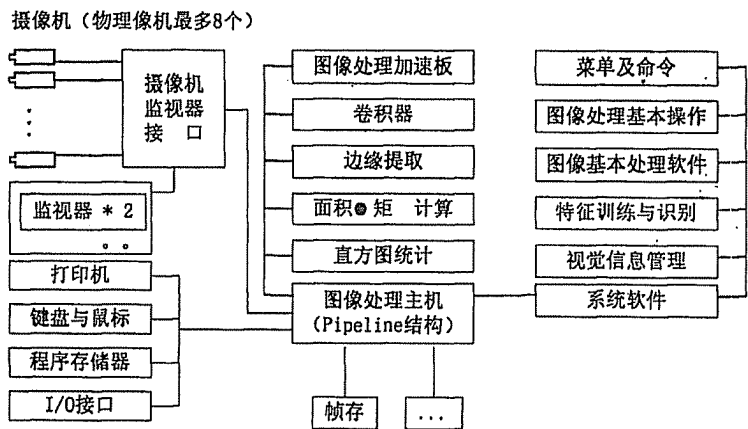


图 3

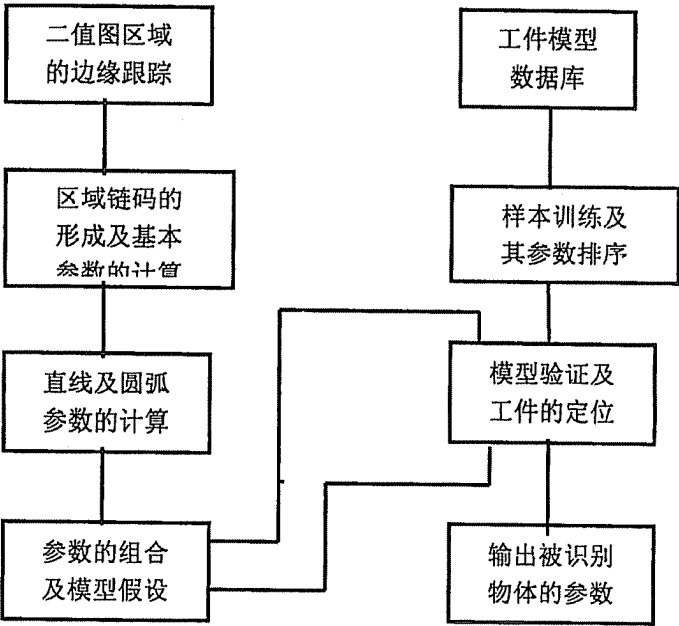


图 3-3

3	2	1
4	P	0
5	6	7

图 3-4

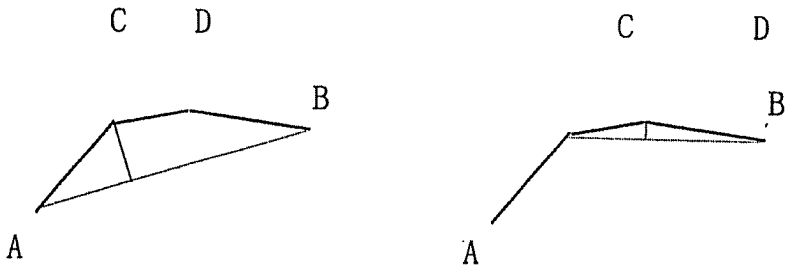


图 3-5

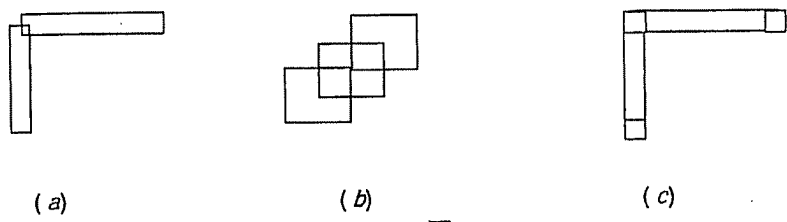


图 3-6

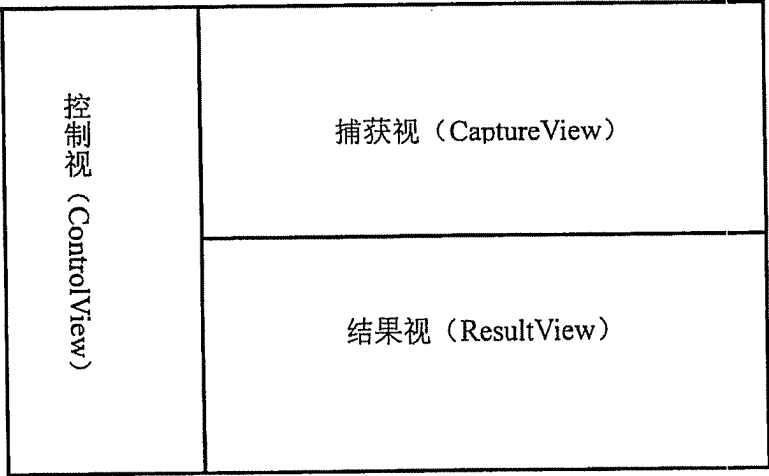


图 3-7

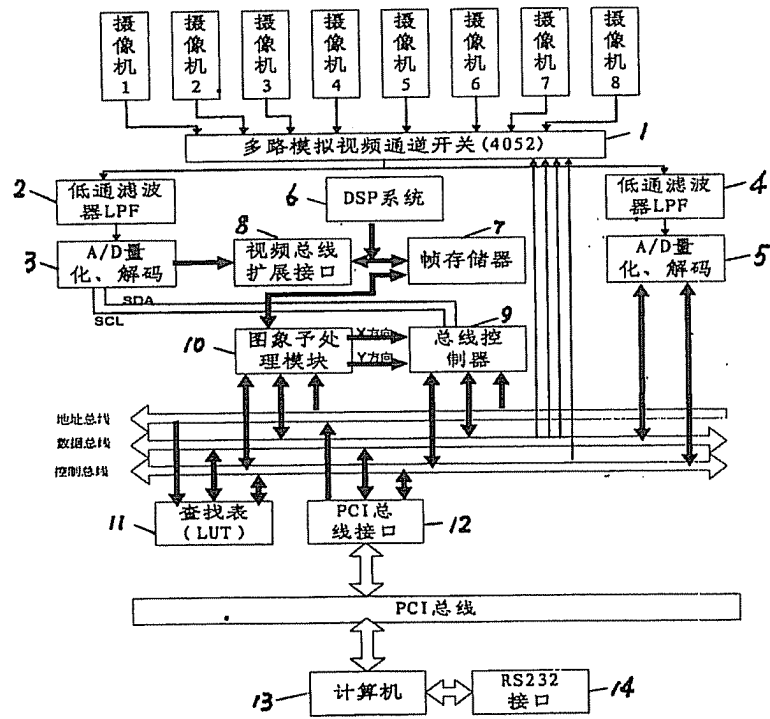


图 4